

FILE SYSTEM WITH PAYMENT FOR ACCESS TO RESOURCES

**FIELD OF THE INVENTION**

The present invention relates generally to computer file systems, and specifically to methods for controlling 5 and monitoring user access to file system resources.

**BACKGROUND OF THE INVENTION**

A file system is a hierarchical collection of files and file directories that are stored on mass storage media, typically on disk, and are accessed using a 10 predefined interface. The term "file system" is also used to denote file system software, i.e., the part of an operating system or an add-on program that supports a file system. The file system software typically includes application program interfaces (APIs) that enable user 15 software applications to access resources in the file system, including specified files, directories and storage volumes. Common file system APIs include, for example, MOUNT(), UNMOUNT(), OPEN(), CLOSE(), READ() and WRITE(). When invoked by a user, these APIs typically 20 generate operating system kernel functions, which operate on the resources specified by the user in the API.

Distributed file systems allow a user on a client computer that is connected to a network to access and modify data stored in files on a file server. When a 25 user accesses data on the file server, a copy of the data may be cached on the client computer, and the user can then read and, if permitted, modify the copy. When the user is finished, the modified data are written back to the file server. Examples of distributed file systems 30 include Sun Microsystems' Network File System (NFS™), Novell Netware™ and the Andrew File System (AFS). Distributed file systems can also be used in accessing

network-attached storage (NAS) devices - hard disk storage that is set up with its own network address, rather than being attached to a server that also serves applications to users on the network, as in classical 5 client/server models.

#### **SUMMARY OF THE INVENTION**

Embodiments of the present invention provide methods, servers and software that enable an owner or manager of a file system to charge users for access to 10 file system resources. These embodiments are based on modifying the conventional file system APIs so as to allow a price to be assigned to each resource in the file system. From the point of view of the user (and of user software applications), however, the semantics and 15 functionality of the APIs are unchanged.

When the user makes an API call to the file system, asking to access a certain resource, the file system software checks the price that has been assigned to the resource. The price may be uniform, or it may 20 alternatively depend on the identity of the user, with different prices assigned to different users or classes of users. The price may be based on substantially any measure of use of the resource, such as duration of use, number of files opened or data volume transferred. The 25 file system software tracks the cost accrued by the user, based on the assigned prices of the resources that the user accesses, and charges the user's account for the accrued cost. Optionally, at least a portion of an amount charged to the user may be credited automatically 30 to suppliers of the resources.

There is therefore provided, in accordance with an embodiment of the present invention, a method for managing a file system, including:

assigning a price to at least one resource of the file system;

receiving an application program interface (API) call from a user of the file system asking to access the 5 at least one resource; and

providing the at least one resource to the user while charging the user in response to the API call for use of the resource in accordance with the assigned price.

10 Typically, the at least one resource includes at least one of a file, a directory and a storage volume.

In one embodiment, assigning the price includes determining charges for use of files in the file system depending on a number of the files opened by the user.

15 In another embodiment, assigning the price includes determining charges for use of the at least one resource depending on a volume of data transferred from the file system to the user. In a further embodiment, assigning the price includes determining charges for use of the at 20 least one resource depending on a length of time during which the at least one resource is in use by the user.

In an aspect of the invention, assigning the price includes assigning different, respective prices for the at least one resource to different classes of users.

25 Typically, receiving the API call includes determining an identity of the user responsively to the API call, and providing the at least one resource includes selecting one of the respective prices depending on the identity of the user.

30 Additionally or alternatively, assigning the price includes assigning different, respective prices to a plurality of different resources.

In exemplary embodiments, receiving the API call includes receiving an invocation of at least one of a mount API, an open API, a read API and a write API.

5 Optionally, the method includes crediting a supplier of the at least one resource with at least a portion of an amount charged to the user for use of the resource.

In a disclosed embodiment, receiving the API call includes receiving a request from a client computer, transmitted over a network to a storage node, to access 10 the at least one resource held in storage by the storage node, and providing the at least one resource includes conveying the at least one resource from the storage node over the network to the client computer.

Typically, charging the user includes at least one 15 of recording a charge against an account held by the user and receiving a transfer of payment from the user.

There is also provided, in accordance with an embodiment of the present invention, apparatus for managing a file system, including:

20 a storage device, which is arranged to store resources of the file system; and

a processor, which is arranged to record a price assigned to at least one resource among the resources stored on the storage device, and which is further 25 arranged, upon receiving an application program interface (API) call from a user of the file system asking to access the at least one resource, to provide the at least one resource from the storage device to the user in response to the API call while charging the user for use 30 of the resource in accordance with the assigned price.

In a disclosed embodiment, the apparatus includes a display, which is arranged to present a listing of the resources so as to enable an operator of the apparatus to assign the price to the at least one resource.

There is additionally provided, in accordance with an embodiment of the present invention, a computer software product, including a computer-readable medium in which program instructions are stored, which 5 instructions, when read by a computer, cause the computer to maintain a record of a price that is assigned to at least one resource of a file system, and upon receiving an application program interface (API) call from a user of the file system asking to access the at least one 10 resource, to provide the at least one resource to the user in response to the API call while charging the user for use of the resource in accordance with the assigned price.

The present invention will be more fully understood 15 from the following detailed description of the embodiments thereof, taken together with the drawings in which:

**BRIEF DESCRIPTION OF THE DRAWINGS**

Fig. 1 is a schematic, pictorial illustration of a system for provision of file system resources, in accordance with an embodiment of the present invention;

5 Fig. 2 is a table that schematically represents a user interface screen that is used in assigning prices to file system resources, in accordance with an embodiment of the present invention; and

10 Fig. 3 is a flow chart that schematically illustrates a method for charging users for access to file system resources, in accordance with an embodiment of the present invention.

#### DETAILED DESCRIPTION OF EMBODIMENTS

Fig. 1 is a schematic, pictorial illustration of a system 20 for provision of file system resources, in accordance with an embodiment of the present invention.

5 In the present embodiment, system 20 is deployed in a facility 22, such as an office building or a hotel, in which multiple users 28, 32, ..., access data resources held at a storage node. In the present example, the storage node comprises disks 26, which are accessed via a  
10 file server 24. This mode of deployment is shown and described here, however, solely by way of illustration. The principles of the present invention may similarly be implemented in substantially any file system, as will be apparent to those skilled in the art.

15 Users 28, 32, ..., operate computer workstations 30, 34, ..., which communicate with file server 24 via a network 36, such as a local area network (LAN). Each user is identified by a username, as is known in the art, which typically identifies not only the user himself/herself, but also the work group and/or organization to which the user belongs. For example, the username JOHN/HAIFA/IBM might identify user "John" as belonging to the Haifa facility of IBM Corporation. Alternatively, in the above-mentioned example in which facility 22 is a  
20 hotel, the username might identify the user's room number for purposes of accessing and being charged for hotel data services provided via network 36.

25 Server 24 maintains data resources on storage volumes 26, which typically comprise hard disks or other mass storage media. These data resources may comprise substantially any sort of content that may be of interest to users 28, 32, .... For example, storage volumes 26 may hold entertainment content, such as movies, music or game programs. Alternatively or additionally, the storage  
30

volumes may hold databases, text files or substantially any other type of information resources known in the art, as well as application programs. A file system on server 24 defines and identifies the volumes, directories and 5 files in which the resources are held. Users 28, 32, ..., access these resources by invoking file system APIs on their respective workstations 30, 34, .... From the point of view of the users and user software applications, the semantics of these APIs may be substantially identical to 10 file system APIs known in the art.

The file system on server 24 includes metadata with respect to each resource held in storage volumes 26. In conventional file systems, these metadata include information such as the file name, directory path and 15 access permissions, indicating which users are authorized to read a given resource and which, if any, are authorized to modify it. The file system on server 24 is extended, in accordance with the present invention, so that the metadata also include price information for some 20 or all of the resources in the file system. When a user asks to access a resource via server 24 by invoking the appropriate API, the server looks up the corresponding price in the metadata, and then charges the user accordingly, in addition to performing the conventional 25 file system functions called for by the API. The accounting of charges may be maintained internally by server 24 or reported to a separate billing server 38 (which charges the user's room account, in the hotel example described above). Optionally, when a resource is 30 provided by a third-party content provider 40, file server 24 may credit the content provider with a portion of the amount charged to users who access the resource. These functions of the file server are described in greater detail hereinbelow.

Typically, server 24 comprises a general-purpose computer processor, which is programmed in software to act as a file server and carry out the functions described herein. The software may be downloaded to the 5 file server in electronic form, over a communication network, for example, or it may alternatively be provided on tangible media, such as CD-ROM. It is a feature of the present invention - although not a prerequisite - that the cost tracking functions described herein can be 10 implemented by straightforward extensions to existing file system software, and without modification to the APIs used by workstations 30, 34, ..., to access the file system. In a sense, these cost tracking functions can be seen as an extension of the access permission function 15 provided by conventional file systems, except that instead of just applying different permission levels to different users (or classes of users), based on their usernames, the file system now applies different access prices to the different classes of users. Denying 20 certain users permission to access a given resource is equivalent to assigning the resource an infinite price with respect to these users.

Fig. 2 is a table that schematically represents a user interface screen 50 displayed by server 24, which 25 enables a file system manager to assign prices to different resources. In the example shown in the figure, the system manager chooses to assign a certain pricing scheme to all files with extension "doc" in directory "xxx" on volume J:. Alternatively or additionally, the 30 manager may choose to apply an individual pricing scheme to certain specific files, or may apply a blanket pricing scheme to an entire directory (such as directory "yyy" in the lower row shown in the figure) or to a group of directories or an entire volume.

The prices assigned by the system manager are differentiated according to classes of users. The charge basis may also be differentiated. Thus, in the present example, users in the Haifa work group of IBM Corporation 5 (\*/HAIFA/IBM) are entitled to free access to the files in question, i.e., COST = \$0.00. Users in other IBM work groups (\*/\*/IBM) are assigned a cost of \$0.10 per file that they access. All other users are assigned a cost of \$0.50 per Megabyte of data that they receive from server 10 24. Alternatively or additionally, other groups of users may be defined, as may other pricing bases, such as a charge based on the length of time during which a given volume is mounted or a given file is open.

In addition, screen 50 includes a "pay to" column, 15 specifying that server 24 is to credit a provider of a certain resource with a portion of the amounts charged to users of the resource. In the present example, this column indicates that GEORGE/HAIFA/IBM is to receive \$0.02 for each file "doc" file in directory J:/xxx that 20 is used by an IBM user, and \$0.20 for each Megabyte of data supplied from these files to other users. This mechanism may be used by the manager of the file system on server 24 to compensate third-party content providers, as noted above. In the present example, it may also be 25 used by the file system owner (in this case IBM) to reward employees who develop useful content. As a further option, the file system on server 24 may allow users to set the prices for their own resources, in a manner similar to the way in which users may set access 30 permissions for their own files in file systems known in the art. For example, users may set prices using a SAVE() API with appropriate extensions, or using a graphical interface similar to screen 50.

Fig. 3 is a flow chart that schematically illustrates a method for charging users for access to a file held in a file system, in accordance with an embodiment of the present invention. The process 5 described here begins, for example, when user 28 of workstation 30 asks to open a particular file on server 24, at a file opening step 60. Prior to step 60, the user may be required to mount the volume on which the file is stored. As noted above, server 24 may 10 additionally or alternatively charge the user for mounting the volume, and not only for accessing files. Details of these mounting functions are omitted here, however, for the sake of simplicity. The methods described hereinbelow with respect to charging users for 15 opening and reading files may be extended in a straightforward way to MOUNT() and other file system APIs.

User 28 (or application software running on workstation 30) initiates step 60 by calling an 20 appropriate file system API, such as OPEN(path/filename). This API causes workstation 30 to send a file request to server 24. The file request specifies the desired path and file and the username of the user requesting the file. Server 24 receives the file request and parses the 25 username to determine the identity of the user and/or the work group or organization to which the user belongs, at a user identification step 62. The server checks this information against the metadata of the requested file. As noted above, the metadata indicate which users are 30 permitted to access the file and if so, at what price.

The server checks the username against the permissions listed in the metadata for the requested file, at a permission checking step 64. If this user is not permitted to access the file, the server returns an

access denial message to workstation 30, at an access denial step 66, as is known in the art. Optionally, the server also checks whether the user is willing and able to pay the specified cost for this file, at a price 5 checking step 68. For example, if the cost of using the file is to be charged against a prepaid or credit account, the server may check that there are sufficient funds or credit remaining in the account to pay for use of the file. Additionally or alternatively, 10 the server may cause workstation 30 to display a message asking user 28 to approve the charge before opening the file. Further additionally or alternatively, when the user views a directory of files on the workstation display, the user may choose to see the prices assigned 15 to each of the files. In any case, if the user is unwilling or unable to pay the price of the file he/she has requested, the OPEN() API exits at step 66, in the manner described above.

Assuming steps 64 and 68 are passed successfully, 20 however, server 24 opens the requested file. Workstation 30 may then read data from the file, typically using a READ() API, at a file reading step 70. Depending on the price basis applicable to user 28, server 24 monitors use 25 of the file, at a monitoring step 72. For example, the server may monitor the number of files, volume of data, or length of time during which files are in use, as illustrated by the different price bases listed in table 50. Based on the use of the file on workstation 30, the server accrues charges against the client's account, 30 until the user closes the file, at a file closing step 74.

The server charges the user for the accrued charges, at a billing step 76. Substantially any suitable payment method known in the art may be used at this step. For

instance, the charges may be billed to a customer account, which is paid periodically, as in telephone billing systems or in the hotel example described above. Alternatively, the charges may be paid immediately using 5 a system of micro-payments or other payment medium to transfer credits over a network. Further alternatively, the charges may simply be recorded in the cost-accounting records of an organization, without any actual exchange of funds. When the file in question contains content 10 supplied by an independent or in-house content provider, as described above, server 24 may also transfer a portion of the charges to the content provider, at a provider payment step 78. This transfer may likewise take the form of either periodic credit, immediate payment, or 15 cost accounting.

Whereas in the embodiment described above, the user is charged in connection with invocation of the READ() API, server 24 may accrue charges based on other file and disk operations, as well. For example, server 24 may 20 charge the user for use of disk space, based on the WRITE() API. The charge in this case could be based on the volume of data written to disk, or on the amount of disk space occupied by data in user files multiplied by the length of time during which the data remain on disk 25 until they are erased. File systems typically have quotas on the amount of disk space allowed to each user, and server 24 may also charge users for the operation of setting the quota.

As noted above, file systems with payment for access 30 to resources may be used in a wide variety of different applications, for example:

- A hotel may collect popular content files, such as music, video and other entertainment content, and may allow guests to connect to the hotel LAN and

access these files for a price. Permissions and prices of the files can be managed as part of the check-in process, with payment at check-out. Different pricing schemes may be applied to 5 different rooms and classes of guests.

- When two organizations, such as business corporations, enter into a partnering relationship, each organization may give the other one access to its intranet. Attaching appropriate prices to the 10 resources on the intranet allows the organizations to build the partnership in such a way that their income and expenditures reflect their actual contributions to the partnership and the benefits they derive.
- An enterprise or other organization can manage its 15 Information Technology (IT) program by charging its own internal customers for their use of file system resources. The accrued charges for different resources indicate to the IT managers which 20 resources are in high demand and which are superfluous. The IT department can thus concentrate its efforts and budget on the actual needs of the organization. The file system charges can also be used to measure employees' use of and contributions 25 to the IT resources of the organization.

Although system 20 is built around a single, central file server 24 on a LAN 36, following the paradigm of distributed file systems, the principles of the present invention may be applied to substantially any sort of 30 storage node, such as NAS devices and nodes in storage area networks (SANS), and to substantially any file system, including parallel file systems. Furthermore, the principles of the present invention are applicable

not only to file systems that serve multiple users, as described above, but also to single-user file systems. In this latter case, the file system may track the use of resources on the local hard disk or optical storage 5 media, for example, and may maintain an accounting record locally and/or transmit credits and debits via a network.

It will thus be appreciated that the embodiments described above are cited by way of example, and that the present invention is not limited to what has been 10 particularly shown and described hereinabove. Rather, the scope of the present invention includes both combinations and subcombinations of the various features described hereinabove, as well as variations and modifications thereof which would occur to persons 15 skilled in the art upon reading the foregoing description and which are not disclosed in the prior art.